WHO’S THERE?

SOFTWARE REQUIREMENT SPECIFICATION

for

Research Practicum 2016

School of computer Science

Document version <1.0>

prepared by

Silvia Saloni, Conor O’Kelly, Ophélie Alliaume and Devin Stacey

Table of Contents

Document revision history

Name	Date	Reason for change	Version

1. Introduction

This report outlines the Software Requirements Specifications (SRS) for developing an automated system estimating room occupancy using Wi-Fi log connection for the school of Computer Science (CS) in the University College of Dublin (UCD).

Wi-Fi coverage is exponentially increasing in our cities and it will be a new form of urban service, such as electricity or phone lines, in the near future (Sevsuck and Ratti 2005). Since Wi-Fi technology was introduced, research tried to take the full advantage of it for getting a better understanding of the human space usage and for applying it to different services, such as: search and rescue, robot explorations, location aware service, smart-health system (Depatla et al. 2015).

Wi-Fi connections turned also to be a good estimate of the human occupancy for public/private spaces and consequently for monitoring human activity in area covered by Wi-Fi signals (Martani et al. 2011), managing smart buildings, optimising services according to the occupancy levels and organising evacuation plans (Depatla et al. 2015).

Those features together with the low costs and the facility of installation make Wi-Fi technology the perfect solution for monitoring the room occupancy of universities (Martani et al. 2011). Other alternatives, like cameras and survey are, in fact, expensive and time consuming (Botta et al. 2015). UCD is spending 25000 euro each year for running a survey for calculating occupancy of the campus rooms.

In this context our web application coupled with a mobile application will be not only useful for cutting UCD costs, but also for optimising the room usage, timetables and rescheduling lecture during the academic year. Furthermore, the information gathered can be potentially being put into use for regulating the energy consume of the building, regulating the flow of students in the campus and helping the coordination of evacuation plans.

IEEE Guide to Software Requirements Specifications (Std 830-1993) were followed for editing the following documentation.

1.1 Purpose

The purpose of this document is to illustrate the implementation and the functionality of the web application together with the mobile application for assessing the occupancy room of the CS building in UCD.

The content of the document will be subjected to change as the project will progress, based on the decisions taken during the costumer meeting and the progress made by the team: Who’s there.

1.2 Scope

The scope of this software is to provide to UCD’s administrative staff a tool capable of collecting all the Wi-Fi log data, store them in a database and fit them to a model for predicting the room occupancy by module and date.

This product will be deployed as a web page viewable within UCD's administrative Ethernet. From the administrator's perspective, users will have access to the data regarding the occupancy of the room for the desired modules or dates, expressed in percentage. Those data are going to be displayed either with the use of a heat map and with graphs by clicking on each room.

The web application will not only be connected to a database, but also to an external server, which contains all relevant data regarding statistics of room occupancy and connections to room's wireless node and all relevant processes involved in the analysis.

The server will also receive data either from external sources to add additional information regarding attendance (e.g. weather, time of day or time of the year) and from a mobile application to get the real estimate of the average number of devices for student within a certain class. In this way the system will optimise the accuracy of the model estimating the room occupancy.

Ideally the use of this system can be extended for regulating the energy consumption in the room and controlling the movement of students in the campus.

1.3 Definitions, Acronyms, and Abbreviations.

CS = Computer Science

UCD = University College of Dublin (UCD)

1.4 References

Botta, F., Moat, H. S., & Preis, T. (2015). Quantifying crowd size with mobile phone and Twitter data. Royal Society open science, 2(5), 150162.

Depatla, S., Muralidharan, A., & Mostofi, Y. (2015). Occupancy estimation using only Wi-Fi power measurements. IEEE Journal on Selected Areas in Communications, 33(7), 1381-1393.

Martani, C., Lee, D., Robinson, P., Britter, R., & Ratti, C. (2012). ENERNET: Studying the dynamic relationship between building occupancy and energy consumption. Energy and Buildings, 47, 584-591.

Sevtsuk, A., & Ratti, C. (2005). iSPOTS. How wireless technology is changing life on the MIT campus. In Proceedings of the 9th International Conference on Computers in Urban Planning and Urban Management, CUPUM.

1.5 Overview

The first part of the document puts in perspective the web application describing its main interfaces, its main functions and its constraints. A section is also dedicated to describe the main characteristics of the users of this system.

In the second part, the functional, the performance and not functional (quality attributes) software requirements are defined in details.

2. The Overall Description

2.1 Product Perspective

This section describes the main interfaces and requirements of the product. This will include a detailed description of the high level interfaces of the software, a rough map for architecture and a look at some of the main user requirements. A more technical review of each section is then carried out in section three.

2.1.1 System Interfaces

This diagram is used to represent a high level look at the layout of the software. The two main user interfaces will be the website and the app, which are managed by a Java server. The server will be also responsible of controlling the database and the data analytics (Figure 1).

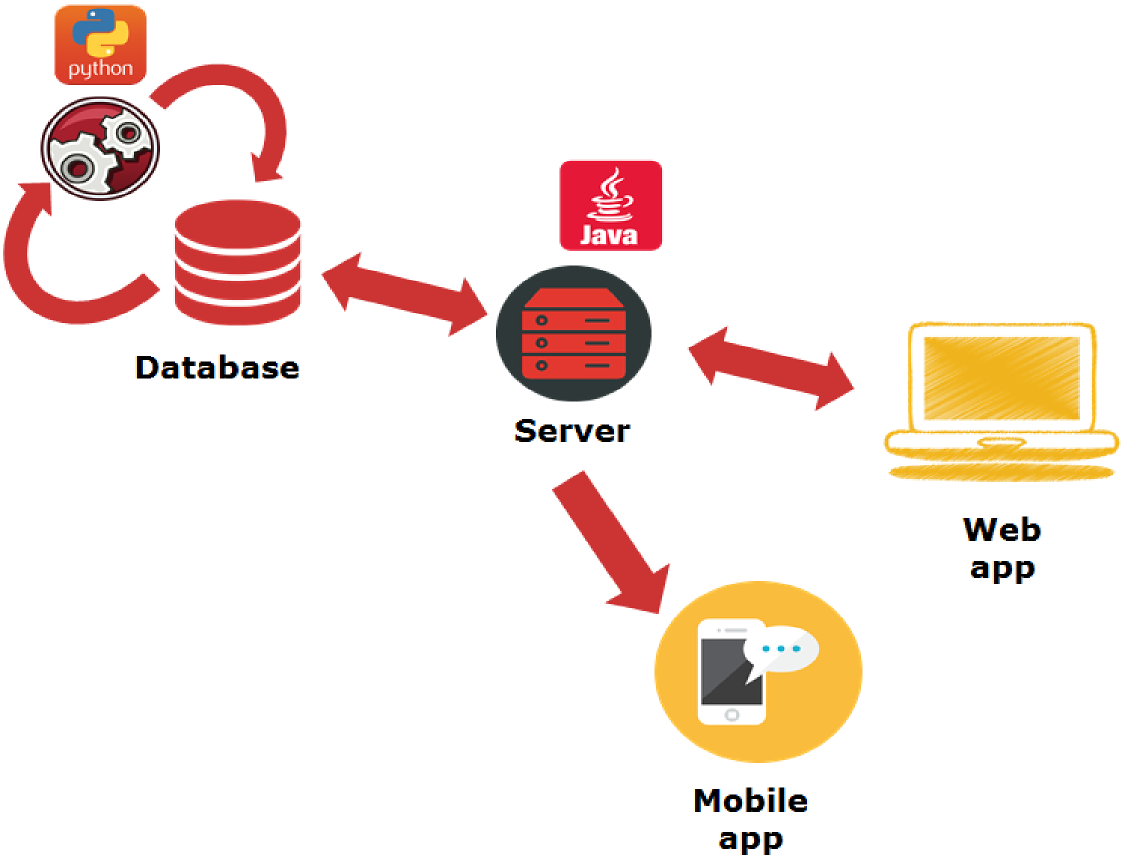

Figure 1 Diagram of the architecture of the system

2.1.2 Interfaces

The system is composed by the different component:

- DB, which stores the data inputs using SQL. Data analyses will be run using Python and the results will be summarised in a table.
- Server will be used to get the request from the user, pull the data from the DB and send them back to the web application. This will be achieved using a Java framework.
- User interface: web app application that allow the user to explore the occupancy of the room
- Mobile application that allow professor to input ground truth data that will be used by the application to correct the prediction of the model

2.1.3 Hardware Interfaces

At this time we do not see the need for any direct hardware interface. Instead the data required will be input through one of the software interface.

2.1.4 Software Interfaces

The system will interact with two interfaces: a web and a mobile application.

The Web interface will be used for two main purposes: the presentation of information based on the current data set and the results of the data analysis that has been carried out on this data set.

The app interface will allow the collection of ground truth data. In particular the user will not only be capable of selecting the room identification number and time of class, but he/she will enter the number of people currently in the room.

2.2 Product Functions

The main function of the project will be the analysis of the provided Wi-Fi data set. This will be achieved using a mathematical model that will be specifically created and tuned for each deployment of the software in the attempt to accurately predict the number of people in a given room. These predictions will be corrected overtime by the use of ground truth data collected through the mobile application. In particular the model will use the ground truth data for re-calculating the ratio of devices connected to Wi-Fi and students. Consequently the model will be more accurate in estimating the number of students.

The outcome of this process will allow the software to present the data via a web interface to the user. This will allow the user to get a high level overview of the occupancy levels of different rooms throughout the semester. This interface will also be able to provide more in depth information if the user requests it.

2.2.1 Data input/ Management requirements

The system should allow the administrator of the web application to change the timetables associated with each room, and it should be able to interface with the data format provided by UCD.

Administrator will be also capable of: adding and removing from the system the rooms under monitoring, changing the meta information presented for each room (e.g. upload new information for a specific room regarding current occupancy and the average number of devices each student possesses), and updating the map used by the system.

Furthermore the system should be capable to accept and upload new Wi-Fi data and ground truth data using a predefined format. The addition of new data can be done also manually for each room.

This category of requirements will allow more recent information on attendance or Wi-Fi connection logs to influence the algorithm to calculate more accurate room occupancy for future predictions.

2.2.2 Data analysis

At this stage the team is still researching on the best model to use to analyse the data provided.

The team, however, agreed on having a model, whose basic ability will be to calculate the number of people in the room based on the number of connected devices. The automated system will utilise a variety of languages in order to accurately analyse a certain data set and show the results to a user.

It will be interesting to break down the classes into different population sets, in order to test whether the different sets will affect the number of people and connected devices ratio.

Furthermore, the model will have the ability to correct overtime the ratio of devices to people through the input of new ground truth data (see above).

As the analysis techniques will not be shown directly to the user, the user will mainly have indirect access to the analysis stage of this piece of software through the input as mentioned in the previous section and the output, as shown in the next section.

2.2.3 Data output

The information obtained from our model is going to be available either in graphical or reports form.

The graphs will be mainly:

- Pie chart showing the current occupancy of the room;
- Line graph showing the daily usage of the room;
- Line graph showing the weekly usage of the room.

The reports will consist of PDF files containing a summary of all the useful information for each room and/or module coupled with the relevant graphs. PDF files will be auto generated by the system and they can be downloaded only for the authorised personnel.

2.2.5 User documentation requirements

The user documentation should have very specific instructions for using any of the provided templates for inputting data into the system. Documentation will also cover methods for interacting with the rest of the system. These guides should be clear and concise.

Templates for general data input will also be provided. This will be tailored to ensure they are very similar to the general way that data will be input into the website or database.

2.3 User Characteristics

There are three main users that will interact with the software and all of them must login into the web application before having access to it.

The general user will be able to access the website and to get a graphical overview of the occupancy of different rooms. The user will have access to all of the historical data.

A Administrator user will be able to access the website and upload new Wi-Fi data or new ground truth data for specific rooms. This type of users will first have to log into the system in order to make sure that they have authorisation. They will then be able to upload new data to the database in a specified format. This user will also be able to download auto generated reports.

The mobile application user will be able to upload new information to the database via the android application. This will involve them selecting the room and time and then submitting a total for the number of people currently present. The information supplied by this type of user will facilitate the tuning of our model.

2.4 Operating Environment

The technicalities of the operating environment are still being discussed at this point with the client. As of the database, data analytics and public server will all take place on a server that has been provided by UCD. This server will act as the connection point for the Web interface and the app.

All of that data analysis will be carried out on this server. At this point in time the team is looking at using a Java controller for the overall server with a full SQL database. In particular, the Java controller that is going to be used is: Spring, since it is the only one capable of working either for building web app and mobile app. The data analytics part of the software will be carried out in Python.

2.5 Apportioning of Requirements

In this section, requirements that may be delayed until future versions of the system are identified, Requirements are divided into different priority for development and delivery.

2.5.1 High priority requirements

Website:

- Selecting buildings and classrooms from a map;
- Searching buildings and classrooms by name;
- Viewing classroom metadata and occupancy data;
- User authentication.

Mobile App:

- Submit ground truth data

2.5.2 Medium priority requirements

Website:

- Search modules by code or name;
- Show classroom and building location on map by module;
- Make available bulk data to be accessed by the user;
- Update classroom metadata;
- Upload ground truth data;
- Upload WI-FI log data;
- Classroom occupancy reports.

Mobile app:

- Access code to authenticate user

2.5.3 Low priority requirements

Website:

- Facility to contact other members of staff for classroom swap;

- Upload timetables;
- Add/remove classrooms;
- Add/remove maps.

3. Specific Requirements

3.1 External Interfaces

3.1.1. Web application

The web application will be used for displaying data on the occupancy of the room in the different building of the campus. The access of this application will be restricted only to UCD staff/students. Therefore, users will need to authenticate themselves with the use of credential. Different kind of users will have different usage of the application. UCD's administrative staff will be the only one that will have the full access of the web application. They are the only one that will be able to upload or modify the data and download the information in form of PDF report.

Students will have access only to the graphical interface, because this might help them: to visualise all the building, where a particular module is taught, check their time table, and see if there are free rooms, where they can go to study in group.

The data displayed are taken directly from a server. The server will be, in fact responsible of taking the data from a database, fit the data into the model and send to the web application the data needed. As stated in section 2.2.3, the data will be presented either as PDF report (only for administrator user) or as graph.

The system should check that the administrator will input data in the proper format and in the right range. If admin user’s input will be incorrect, an error window should pop up and warn the user of the mistake. Ideally the system should also warn the user that she/he is deleting data regarding certain class or module, in order to prevent that the user will inadvertently delete important information from the web application.

Error message should appear also if the web application is not communicating properly with the Server.

The communication with the server should be fast and the user should not experience delay in the viewing of the information is interested.

The main home page of the web application is shown in the prototype below (Figure 2).

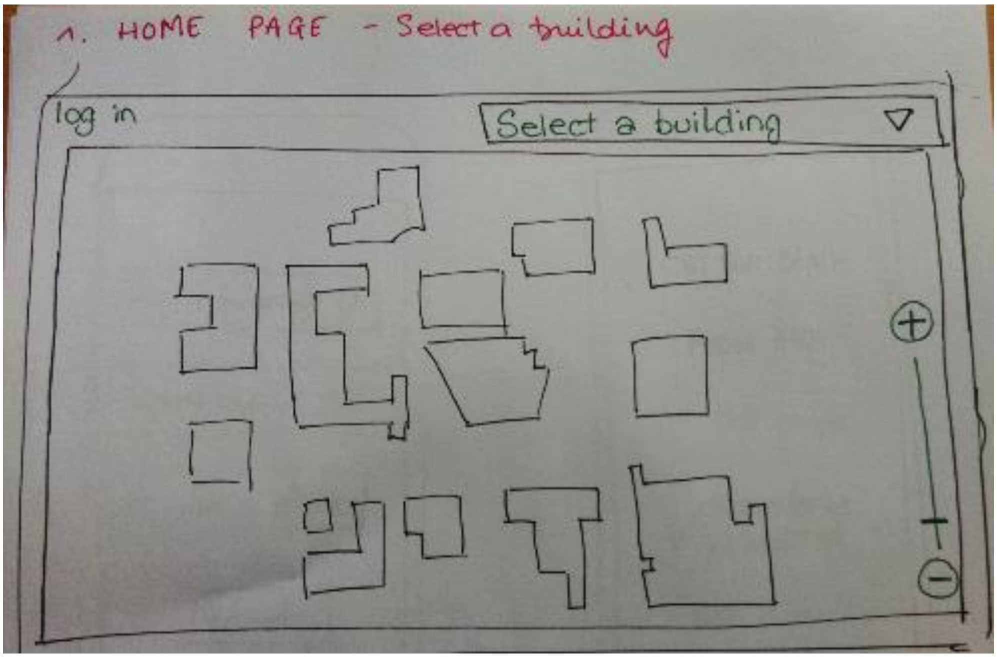

Figure 2 Prototype for the home page for the web application

The user will be able to access this section via UCD's Administrative Ethernet. From there, a user will be able to select from a drop down menu a specific room or module. Alongside this, the user has the possibility to select directly a building using the UCD map.

If the user will select a building, it will go in a page where he/she will have the possibility of selecting a room either using the dropdown menu or clicking on the room in the map (Figure 3).

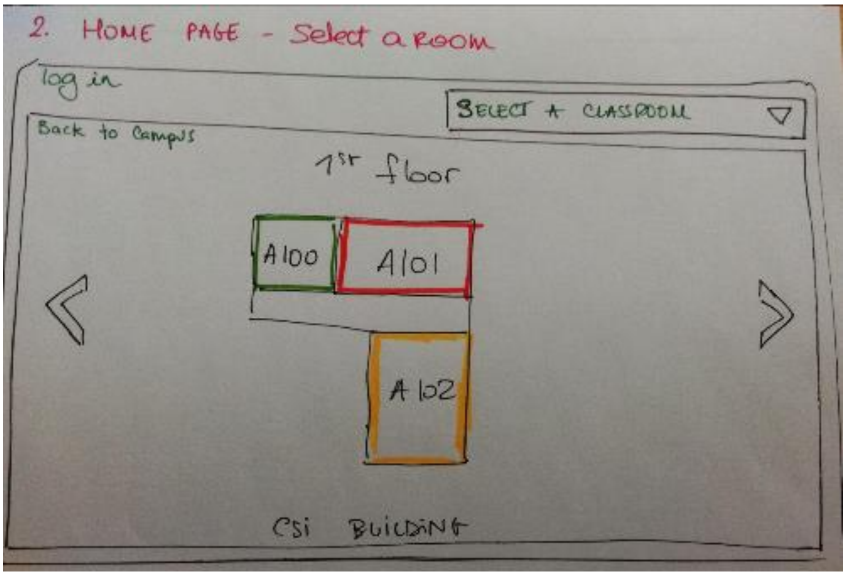

Figure 3 Prototype of the page displaying the building map

Once the user has selected a particular room he/she will be directed to the room page, where the main information are going to be graphically displayed as follow (Figure 4):

- Pie chart showing the current occupancy of the room;
- Line graph showing the weekly usage of the room;
- The module that is at the moment on in the classroom
- The main features of the room in term of capacity, plugs ect.

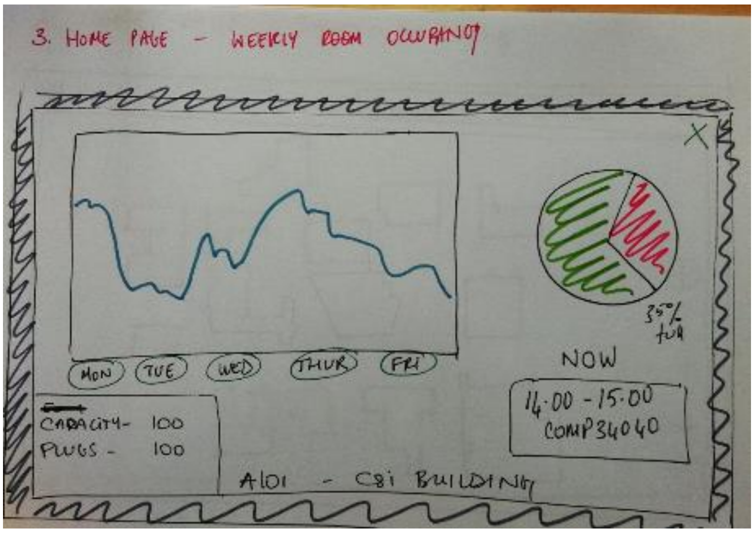

Figure 4 Prototype of the web page showing the main information for each classroom.

The user will have access to the daily usage of the room by click on each day in the labels of the axis of the graph.

On the page displaying the daily usage of the room, the user can see the occupancy of the room and the modules for each hour (Figure 5).

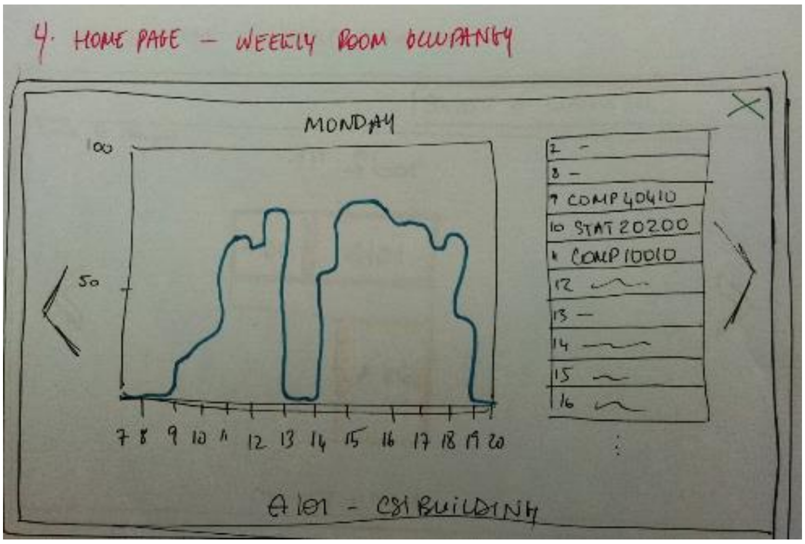

Figure 5 Prototype of the web page showing the daily usage of the room.

In this page the user will be able to click on the module and see in which building the module is taught during all the week (Figure 6). This feature will help student in organising his/her movement in the campus especially in the first month of the semester.

This page can be reach directly selecting the module from the drop down menu in the main home page (Figure 2).

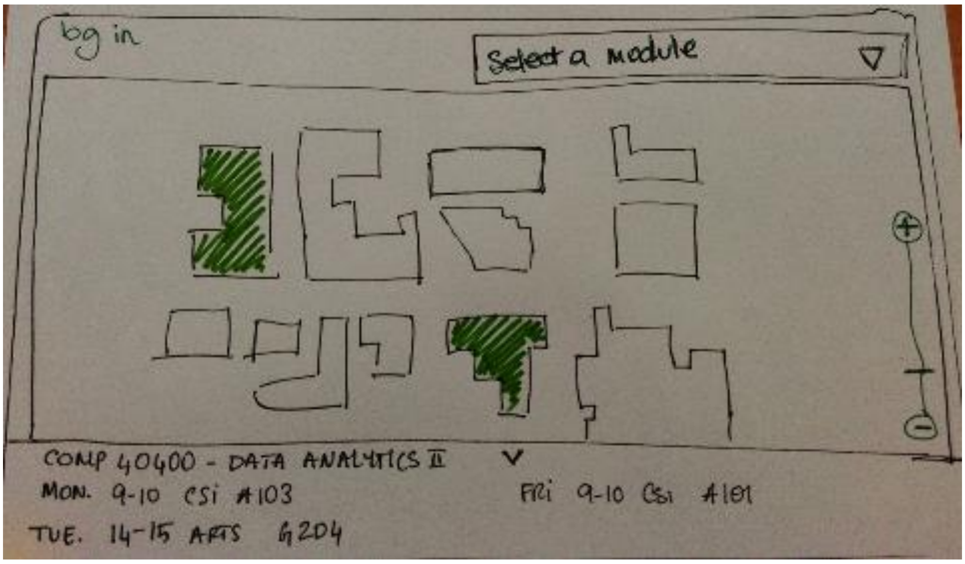

Figure 6 Prototype of the web page showing the weekly usage of the room.

3.1.2. Mobile application

The mobile application will be used for collecting the ground truth data, which will be used by the model for correcting its predictions.

The data will be inputted for each lecture held in the class by the lecture by mean the mobile application. Once the data are collected, they will be sent by the app to the server, which will send them to the database.

The data before being sent to the server needs to be checked. If the data are in the right format, the data will be sent and a message for the successful submission will appear on the mobile screen (Figure 7). If not, an error message will warn the user that the format of their input was incorrect. An error message will occur also if the data are not sent to the server.

The application form should be also fault tolerant towards hacker attack. An error message should appear if the form will be completed with non-numeral format or with malicious intent.

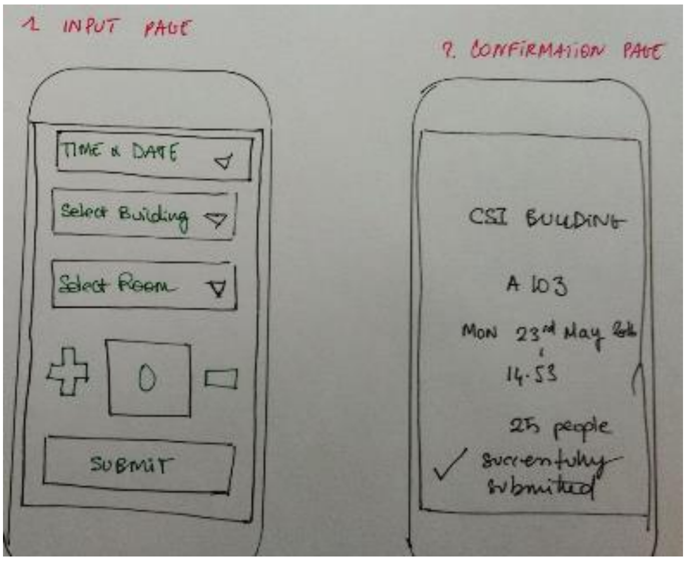

Figure 7 Prototype of the mobile app data input page.

The information should be sent quickly to the server, if not an error message will appear. For this reasons, our team decided to use Java controller Spring. Java controller usually guarantees a good speed communication the database and the application. Although Spring is heavier than other program, it was the only one capable to communicate with an Android application.

An initial prototype is shown in the figure below on how the mobile application should look like (Figure 7).

Thanks to this page the user will choose the date, the module, the room and she/he be able to input the number of students in the class. The total number of students can be directly inputted typing the number or using the plus and minus button and use the app as a manual counter.

The format with which the data are going to organised has still to be decided by the team.

3.2 Functions

In this sections there are going to be listed the functionality that the team thinks to be necessary for having a complete system for estimating the room occupancy of UCD room.

All the two users' interfaces of the system can be only used by UCD staff/students to preserve the privacy of the information. The website and the mobile application will be accessible only through a password authentication. The registration process should give an option to create either a student account or a professor account to ensure the separation in the functionality of their workspaces.

3.2.1 Website functions

3.2.1.a General user functionality

- The system shall provide user request page for getting the password;
- The system shall provide a login page and make sure that user credentials are correct. If not the access should be denied;
- The system shall provide a user profile page, where he/she can update his/her personal information and password.
- The system shall allow the user to have access to the map of the computer science building in UCD for the moment. In the future the user will have access to all the building in UCD;

- The system shall allow the user to see the map of the building for seeing the location of rooms within the building;
- The system shall show the occupancy of the room based on the colour of the room in the map;
- The system shall allow the user to select the room (for which the occupancy data are available) to see more information on the room either directly clicking on the room directly or selecting it from a drop down menu;
- The system shall provide to the user the results of the data analysis available in term of occupancy or the current moment, the day and the week using graphs;

3.2.1.b Administrator functionality

- The system shall provide to the administrator the same functionality offered to the general user;
- The system shall provide to the user the results of the data analysis available in PDF form as discussed in the previous section;
- The system shall give to the administrator the ability to change the timetable of the rooms;
- The system shall make sure that the administrator will input the data in the format desired by the UCD;
- The system shall allow the administrator to add and/or remove what rooms are being monitored;
- The system shall allow the administrator to add and/or remove maps of the building;
- The system shall give the rights to the administrator to change meta information about rooms;
- The system shall permit the administrator to upload new Wi-Fi data in a specified format;
- The system shall give the ability to the administrator to manually add ground truth data for each room through the website using a specified format.

3.2.2 Mobile application functions

- The system shall provide a login page and make sure that user credentials are correct. If not the access should be denied;
- The system shall allow the users to select the room, time and module for which he/she want to input ground truth data;
- The system shall checked if the data inputted are in the correct format and do not have any malicious intent;
- The system shall inform the user if the data input was not correct or there was a problem in the submission of the data;
- The system shall inform the user if the data input operation went correctly.

3.2.3 Data Analysis functions

The team has still to define the final functionality for the data analysis. At the moment the team agreed that the system shall:

- Have a model that calculate the number of people in the room with a good accuracy;
- Have a model that use the ground truth data inputted using a mobile application for recalculate the average of people over time and improve the accuracy of the model.

3.2.4 Data management functions

3.2.4.1 Data input of templates into the website

Admin will be able to upload template files directly to the website via a drop interface. These files will then be processed by the server, verified and then added to the database.

3.2.4.2 API for bulk data download

Api will be available for authorised users to download the bulk data set. The queries to the DB will include search terms to allow users to selectively access information. The current proposed format of this download will be in JSON.

3.2.4.3 Auto generated PDF reports

The user will be able to download an automatically generated PDF report from the site. This can be customised by selected specific rooms / buildings / modules / time period.

3.2.5 Data cleaning for input functions

The data need to be clean and formatted according to the specified documentation. For this reason a series of functions were created for cleaning the given dataset. This small bridging script will be unique to each location. Ideally a series of data input templates will be provided to administrators for organising the data using the same format of the tables used in the database. This will allow to bypass the cleaning step and to input the data directly into the dataset.

3.2.6 Documentation functions

This section details the templates that should be provided for users in order to input data into the database via the website.

3.2.6.1 Template for database data input / modification

Standard Excel template will show in details how information should be laid out in the file. This template can be filled out by the relevant admin staff and used to submit new data to the database.

3.2.6.2 Template for WI-FI date input

Basic templates based on the current file layout that we have gotten from the customer.

3.2.6.3 Template for adding in bulk ground truth data

Excel file template for adding bulk ground truth data to the database.

3.2.7 Future software functions

3.2.7.1 Ability for users to changes map / rooms format

On site software admin staff can change the layout of maps / rooms via an interactive GUI. This will allow new updates to be added to maps to reflect changes in the layout of buildings.

This will allow admin to change map / room layout on site without needing interaction with the development team.

3.2.7.2 Allows users to request a room swap

Users of the website will be able to request a swap of room with another user's. This feature will allow the users to specify the two room numbers as well as the time that they would like swapped.

The request will be generated by filling out a form, which will be sent as email. This email will show the credential of the individuals and the classrooms involved in the swap together with the relevant timetabling management.

In this way users will take advantage of underutilized or empty rooms.

3.3 Performance Requirements

This web-based application should have the necessary storage space needed to store the data pertaining to all UCD building and classroom locations and metadata, classroom historical and predicted occupancy and user account information. It should also be capable of running flawlessly with at least 10 people.

Checking classroom occupancy should be fast and the data should not take more than a couple of seconds to upload. This application should also run smoothly even with slow internet connection.

3.4 Logical Database Requirements

In order to design the database to fulfil our needs, we shall require at least three tables outlying the most important information about our data. The data we will require is split into three separate sections; Module, Lecture and Room. This will allow us to facilitate any features that will be required by our software and each table will be outlined in detail below.

3.4.1 Module

In order to provide information regarding occupancy for individual modules, there will be a specific table detailing the information regarding a certain module. The information here will include a Module Code (Primary Key), a Year and the School the module is associated with. This table will allow us to provide full information regarding a module to allow us to prepare infographics regarding these modules.

3.4.2 Lecture

This table will provide full information regarding a specific lecture timeslot. The columns found within this table will be named Time and Room Name (Both Primary Keys), Module Code, Associated Count, Authenticated Count and Occupancy. This table will provide the bulk of the information that our software will use. The associated and authenticated count will provide information regarding how many connections were made to the local wireless node during the time period shown within the Time column. Equally, the occupancy column will show a percentage representing how full the lecture was to the nearest 25%. This table will be connected to the external tables of Module via the Module Code foreign key and to the Room Table via the Room name foreign Key.

3.4.3 Room

This table will provide full information regarding a specific room. The columns found within will be Name and Building (Primary Keys) alongside Capacity. This information will be used to provide in depth statistics regarding the occupancy of individual rooms, alongside a room’s address outside of the map.

3.5 Design Constraints

The input file format used by this software will be standardised as a CSV file or xlsx extensions in order to fit in with the standard set by the University. Any different file formats needed to be implemented due to future changes to the standard by the University will be implemented at a later stage.

The output format of our information, aside from the information provided by our in built web page, shall be in PDF format showing the infographics requested by the user.

As the system includes both a system for a web application and a supporting Android application, for ease of development, we'd encourage utilising supporting layers that are compatible between the two

3.6 Software System Attributes

3.6.1 Reliability

It is important that the system should not crash while inputting occupancy data with the mobile application. The system should also be able to handle csv files uploaded by the user that do not match the required format without impeding on its correct running.

3.6.2 Availability

The system should be available at all time and should not prevent users from submitting classroom occupancy data with the mobile app or through the website. Users should also be able to view classroom occupancy data at any time.

3.6.3 Security

The web application should only be accessed by registered users who have been authenticated. User accounts should not be accessed or modified by other users. The system should be secure so that it is immune to attacks.

All data relating to users, module timetables and classroom occupancy should not be shared with or modified by registered and non-registered users. Only registered admin users should be able to edit classroom metadata.

Mobile app users should only be able to submit classroom occupancy data after providing an access code.

3.6.4 Maintainability and extensibility

The system should be built in a modular way in order to facilitate the addition of new functionality and easy testing and debugging.

The system should be well documented in order for developers who might not be familiar with the system to be able to understand it and modify it.

3.6.5 Portability

The web application should work correctly and its user interface should look the same on different browsers.

The user interface of the mobile app should also look similar on different screen sizes.

4. Change Management Process

The team will follow an Agile methodology for realising this application. Therefore, all the requirements that need to be achieved will be broken down in smaller tasks, which need to be completed at the end of each sprint. At the end of each sprint the team aims to provide to the customer a prototype to be tested. This will allow the team to check if the product met the desire of the customer. If the product satisfies the customers, the team will continue with the predefined plan. In the opposite scenario, the team will take on board the customers' critics or suggestion and implement them in the next sprint.

Following this methodology will ensure the team to make changes as the project will progress.

5. Document Approvals

The document is approved by the following customer panel:

Professor Padraig Cunningham

Dr. Julie Berndsen

Dr. Gavin McArdle

Dr. Anthony Ventresque

date: ___ /___ /___